

# HCDR3 Loop Modeling

**Bold text means that these files and/or this information is provided.**

*Italicized text means that this material will NOT be conducted during the workshop.*

Fixed width text means you should type the command into your terminal.

If you want to try making files that already exist (e.g., input files), write them to a different directory!

This tutorial assumes that you have Rosetta added to your PATH variable. If you do not already have this done, add the rosetta applications to your path. For the Meilerlab workshop (tcsh shell), do this:

```
setenv PATH ${PATH}:${HOME}/rosetta_workshop/rosetta/main/source/bin
setenv PATH ${PATH}:${HOME}/rosetta_workshop/rosetta/main/source/tools
```

alternatively, for bash shell users:

```
export PATH=${HOME}/rosetta_workshop/rosetta/main/source/bin:$PATH
export PATH=${HOME}/rosetta_workshop/rosetta/main/source/tools:$PATH
```

## Tutorial

This tutorial presents an HCDR3 loop modeling benchmark experiment. The human monoclonal antibody 5J8 binds and neutralizes a broad spectrum of modern H1N1 influenza viruses including the 2009 pandemic H1N1 virus (Krause et al., J Virol 2011). The antibody was crystallized at a resolution of 1.55 angstroms and the resulting structure was submitted to the Protein Data Bank (PDB) under accession number 4M5Y. In this tutorial, you will reconstruct the HCDR3 loop of 5J8 using *de novo* loop modeling. At the end of the tutorial, the results from this benchmark experiment will be compared to the native structure available from the PDB.

1. Prepare your working directory. You will work in this directory for the rest of the tutorial.
  1. Create a directory in the hcdr3\_modeling directory called my\_files and switch to that directory.

```
mkdir my_files
cd my_files
```

2. Prepare your input files.

1. Prepare a PDB input file. Typically, this is accomplished by removing unnecessary chains, waters and non-protein molecules (e.g. gold) leaving behind one asymmetric unit containing one V domain of a heavy and light chain pair. Cleaned PDB files are then renumbered using the renumber\_pdb.py script.

**The clean and renumbered 4m5y\_renum.pdb file is provided for you in the input\_files/ directory.**

1. Get the native PDB structure.
  1. Go to <http://www.pdb.org/>.
  2. Search for 4M5Y.
  3. Click "Download Files" and "PDB File" in the top right hand corner by the PDB ID.
  4. Save the PDB file as 4m5y.pdb.
  5. The file may end up downloading to your Downloads directory. If so, move it to your working directory using this command (don't forget to include the "." at the end of the command):

```
mv ~/Downloads/4m5y.pdb .
```

2. Using a text editor (such as vi or gedit) or Pymol, clean 4m5y.pdb so that only the V domain of chains H and L remain. Manually delete residues 115-213 on chain H and residues 107-209 on chain L. Remove all other chains and any HETATM lines. Save the truncated file as 4m5y.pdb

```
gedit 4m5y.pdb
```

3. Renumber 4m5y.pdb:

```
python ../scripts/renumber_pdb.py
-p 4m5y.pdb -o 4m5y_renum.pdb
```

2. Generate a FASTA sequence file from this PDB file.

**The 4m5y\_.fasta file is provided for you in the input\_files directory.**

1. Run the following script to save the sequence as 4m5y\_.fasta in your working directory:

```
../scripts/get_fasta_from_pdb.py \
4m5y_renum.pdb H 4m5y_.fasta
```

3. Generate a loops file.

**The 4m5y\_.loops file is provided for you in the input\_files directory.**

1. Create a file containing one line formatted as follows: LOOP [residue before HCDR3 loop begins] [residue after HCDR3 loop ends] 0 0 0

```
LOOP 96 114 0 0 0
```

4. Prepare 3mer and 9mer fragment libraries.

**The 4M5Y fragment libraries (4m5y\_fragments.200.3mers and 4m5y\_fragments.200.9mers), secondary structure prediction (4m5y\_.psipred\_ss2 and 4m5y\_.jufo\_ss), checkpoint (4m5y\_.checkpoint) and homolog exclusion (4m5y\_.homolog\_vall) files are provided for you in the input\_files directory.**

1. Make fragment library files using Robetta (for the purposes of this workshop).

1. If you are an academic or non-profit user of Rosetta, make sure you're registered at <http://rosetta.bakerlab.org/>.
2. Under "Services", "Fragment Libraries" click "Submit".
3. Type 4m5y\_ under "Target Name".
4. Copy/paste all the text in 4m5y\_.fasta into the "Paste Fasta" field.
5. If you are benchmarking, you will want to check "Exclude Homologues".
6. Click "Submit".
7. You can see your position in the queue by clicking "Queue" under "Fragment Libraries". This should not take very much time unless the queue is long. For time considerations, you may use the provided files.
8. When your job is finished, you should receive an email with a link called "Result Details". Click on this link. Then, click on "DOWNLOADS".
9. Download the checkpoint, psidpred\_ss2, jufo\_ss and homolog\_vall files. Right click on the files, and save the files to your working directory.
10. If copying the provided files, use the following commands (do not forget to include the "."):

```
cp ../input_files/4m5y_.psipred_ss2 .
cp ../input_files/4m5y_.jufo_ss .
cp ../input_files/4m5y_.checkpoint .
cp ../input_files/4m5y_.homolog_vall .
```

2. Prepare the fragment picker setup files.

**The fragment\_picker\_quota.options, fragment\_picker\_quota.wghts and fragment\_quota\_picker.cfg files are provided for you in the input\_files directory.**

1. Prepare the fragment picker options file. Due to time considerations, these files have been prepared for you. Below are some considerations for when making a fragment picker options file:

- Rosetta ignores text beginning with # (these are treated as comments).
- Avoid mixing tabs and spaces. Be consistent in your formatting (tab-delimited or colon-separated).

2. Save the options, config and weights file in your working directory:

```
cp ../input_files/fragment_picker_quota.* .
```

3. Run the fragment picker.

1. Make sure all of the filenames and paths in the options file are correct.
2. Make sure you are in your working directory.
3. Type the following command, replace ROSETTA with the path to your local rosetta installation, for the MeilerLab workshop, this would be \${HOME}/rosetta\_workshop/rosetta/:

```
fragment_picker.default.linuxgccrelease @fragment_picker_quota.options \
-in:file:vall ${HOME}/rosetta/workshop/rosetta/tools/fragment_tools/vall.jul19.2011.gz
```

5. Optional: Prepare HCDR3 torso restraints. These restraints improve modeling of antibodies with bulged torso configurations (North et al., J Mol Bio 2011).

**The 4m5y\_bulged.restraints file and the dihedral\_cst.wts\_patch are provided for you in the input\_files directory.**

1. Prepare a restraints file. A script has been provided to make restraints file formatting easy, however these files can be manually created.

```
../scripts/maketorsoconstraints.py -b -s 97 -e 113 > 4m5y_bulged.restraints
```

2. Prepare a weights patch file named dihedral\_cst.wts\_patch containing the following line to turn on dihedral angle constraint scoring.

```
dihedral_constraint 1.0
```

6. Prepare the loop modeling options file.

**The model\_w\_rest.options and model\_wo\_rest.options files are provided for you in the input\_files directory.**

1. Depending on whether or not you choose to model your antibodies with HCDR3 torso restraints, prepare the loop modeling options file.

1. Rosetta ignores text beginning with # (these are treated as comments).
2. Avoid mixing tabs and spaces. Be consistent in your formatting (tab-delimited or colon-separated).
3. The options file for modeling with restraints includes additional flags for restraint file handling.

```
cp ../input_files/model_w_rest.options .
cp ../input_files/model_wo_rest.options .
```

3. Run Rosetta LoopModel.

1. Make sure all the filenames and paths in the options file are correct.
2. Make sure you are in your working directory.
3. Type the following command line to model the HCDR3 loop with restraints:

```
loopmodel.default.linuxgccrelease @model_w_rest.options -out:prefix w_rest-
```

4. Type the following command line to model the HCDR3 loop without restraints:

```
loopmodel.default.linuxgccrelease @model_wo_rest.options -out:prefix wo_rest-
```

- NOTE: This will take ~30 minutes per structure. Please move forward in the tutorial using the pre-created models and refer back to the models you generate at a later time.

#### 4. Analyze your data

**Example data is provided for you in the**

*\*output\_files/example\_data/*

**directory.**

1. For practice, we will be analyzing data that has already been generated.

1. Create a directory for analysis of your data and switch into that directory.

```
mkdir data_analysis
cd data_analysis
```

2. Copy the .pdb files from the output\_files/example\_data/ directory.

```
cp ../../output_files/example_data/*.pdb .
```

3. Copy the native, renumbered 4M5Y structure from the input\_files/ directory.

```
cp ../../input_files/4m5y_renum.pdb .
```

4. Create a resfile defining the residues of the HCDR3 loop. For the purposes of this tutorial, the resfile has been created for you. Copy it to your directory and review its contents.

```
cp ../../output_files/4m5y.resfile .
gedit 4m5y.resfile
```

5. Run the score\_vs\_rmsd\_byres.py script to generate a score vs. RMSD table.

**For this script to run, replace Rosetta with the path to your rosetta installation, e.g. `${HOME}/rosetta_workshop/rosetta`. If necessary, open the script and define `clustalw_exe` with your local clustalw executable**

```
~/rosetta/workshop/rosetta/tools/protein_tools/scripts/score_vs_rmsd_full.py \
-n 4m5y_renum.pdb -o SVR -r 4m5y.resfile cons-*.pdb
```

1. Review the output tables, and generate plots of total score vs. CA RMSD using your favourite graphing software.

```
gedit SVR_align_all_model.tsv
gedit SVR_align_by_residue.tsv
```

2. Visualize the example models in PyMol. Compare your observations with the score vs. RMSD data.

```
pymol *.pdb
```